**IN THE CLAIMS**

For the convenience of the Examiner, all pending claims of the present Application are shown below in numerical order whether or not an amendment has been made.

1-39. **(Canceled)**

40. **(Previously Presented)** A method of generating dependency information for code objects stored in a database, comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database;

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects; and

stopping the recursive query of the database upon identifying a dependency that is already included in the dependency information tracking array.

41. **(Previously Presented)** The method of Claim 40, further comprising:

recursively querying the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identifying one or more dependencies of specifications of object-oriented code objects stored in the database; and

incorporating the one or more dependencies of specifications of object-oriented code objects into the dependency information tracking array.

42. **(Previously Presented)** The method of Claim 40, further comprising:

recursively querying the database for one or more dependencies of implementations of object-oriented code objects in the database;

identifying one or more dependencies of implementations of object-oriented code objects in the database; and

incorporating the one or more dependencies of implementations of object-oriented code objects in the database into the dependency information tracking array.

43. **(Previously Presented)** The method of Claim 40, further comprising:

parsing the source code of the database for data manipulation statements that fire triggers; and

identifying one or more data manipulation statements that fire triggers.

44. **(Previously Presented)** The method of Claim 43, wherein parsing the source code of the database for data manipulation statements that fire triggers comprises parsing the source code for UPDATE, DELETE, or INSERT statements.

45. **(Previously Presented)** The method of Claim 43, further comprising:

recursively querying the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identifying one or more dependencies on triggers of code objects stored in the database; and

incorporating the one or more dependencies on triggers of code objects stored in the database into the dependency information tracking array.

46. **(Previously Presented)** The method of Claim 40, further comprising compiling one or more code objects stored in the database in debug mode using a database code object debugging tool.

47. **(Previously Presented)** The method of Claim 40, further comprising identifying one or more dependent objects stored in the database that are INVALID.

48. **(Previously Presented)** A method of generating dependency information for code objects stored in a database, comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database;

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects; and

identifying one or more cyclic dependencies among code objects stored in the database.

49. **(Currently Amended)** The method of ~~**Claim 75,**~~ **Claim 48,** wherein identifying one or more cyclic dependencies comprises utilizing a graph traversal algorithm to identify one or more cyclic dependencies.

50. **(Previously Presented)** The method of Claim 40, further comprising generating a dependency graph for code objects stored in the database based at least in part on the dependency information tracking array.

51. **(Previously Presented)** The method of Claim 40, wherein the database comprises a database catalog; and

wherein querying the database comprises querying the database catalog.

52. **(Previously Presented)** A method of generating dependency information for code objects stored in a database, comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database;

recursively querying the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identifying one or more dependencies of specifications of object-oriented code objects stored in the database;

recursively querying the database for one or more dependencies of implementations of object-oriented code objects in the database;

identifying one or more dependencies of implementations of object-oriented code objects in the database;

parsing the source code of the database for data manipulation statements that fire triggers;

identifying one or more data manipulation statements that fire triggers;

recursively querying the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identifying one or more dependencies on triggers of code objects stored in the database; and

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects stored in the database, one or more dependencies of specifications of object-oriented code objects stored in the database, one or more dependencies of implementations of object-oriented code objects stored in the database, and one or more dependencies on triggers of code objects stored in the database.

53. **(Previously Presented)** The method of Claim 52, wherein parsing the source code of the database for data manipulation statements that fire triggers comprises parsing the source code for UPDATE, DELETE, or INSERT statements.

54. **(Previously Presented)** The method of Claim 52, further comprising compiling one or more code objects in debug mode using a database code object debugging tool.

55. **(Previously Presented)** The method of Claim 52, further comprising identifying one or more dependent objects in the database that are INVALID.

56. **(Previously Presented)** The method of Claim 52, further comprising identifying one or more cyclic dependencies among code objects stored in the database.

57. **(Previously Presented)** The method of Claim 56, wherein identifying one or more cyclic dependencies comprises utilizing a graph traversal algorithm to identify one ore more cyclic dependencies.

58. **(Previously Presented)** The method of Claim 52, further comprising generating a dependency graph for the code object based at least in part on the dependency information tracking array.

59. **(Previously Presented)** The method of Claim 52, wherein the database comprises a database catalog; and

wherein querying the database comprises querying the database catalog.

60. **(Previously Presented)** A system for generating dependency information for code objects stored in a database, comprising:

a database operable to store code objects; and

a processing system operable to:

recursively query the database for one or more dependencies of procedural code objects stored in the database;

identify one or more dependencies of procedural code objects stored in the database;

generate a dependency information tracking array based on the identification of one or more dependencies of procedural code objects; and

stop the recursive query of the database upon identifying a dependency that is already included in the dependency information tracking array.

61. **(Previously Presented)** The system of Claim 60, wherein the processing system is further operable to:

recursively query the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identify one or more dependencies of specifications of object-oriented code objects stored in the database; and

incorporate the one or more dependencies of dependencies of specifications of object-oriented code objects stored in the database into the dependency information tracking array.

62. **(Previously Presented)** The system of Claim 60, wherein the processing system is further operable to:

recursively query the database for one or more dependencies of implementations of object-oriented code objects stored in the database;

identify one or more dependencies of implementations of object-oriented code objects stored in the database; and

incorporate the one or more dependencies of implementations of object-oriented code objects stored in the database into the dependency information tracking array.

63. **(Previously Presented)** The system of Claim 60, wherein the processing system is further operable to:

parse the source code of the database for data manipulation statements that fire triggers;

identify one or more data manipulation statements that fire triggers;

recursively query the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identify one or more dependencies on triggers of code objects stored in the database; and

incorporate the one or more dependencies on triggers of code objects stored in the database into the dependency information tracking array.

64. **(Previously Presented)** A system for generating dependency information for a code object stored in a database, comprising:

a database operable to store a code object; and

a processing system operable to:

recursively query the database for one or more dependencies of procedural code objects stored in the database;

identify one or more dependencies of procedural code objects stored in the database;

recursively query the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identify one or more dependencies of specifications of object-orients code objects stored in the database;

recursively query the database for one or more dependencies of implementations of object-oriented code objects stored in the database;

identify one or more dependencies of implementations of object-oriented code objects stored in the database;

parse the source code of the database for data manipulation statements that fire triggers;

identify one or more data manipulation statements that fire triggers;

recursively query the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identify one or more dependencies on triggers of code objects stored in the database; and

generate a dependency information tracking array based on the identification of one or more dependencies of procedural code objects stored in the database, one or more dependencies of specifications of object-oriented code objects stored in the database, one or more dependencies of implementations of object-oriented code objects stored in the database, and one or more dependencies on triggers of code objects stored in the database.

65. **(Previously Presented)** A computer-readable medium encoded with logic operable, when executed on a computer processor, to perform the steps comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database; and

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects; and

stopping the recursive query of the database upon identifying a dependency that is already included in the dependency information tracking array.

66. **(Previously Presented)** The computer-readable medium encoded with logic of Claim 65, further operable to perform the steps comprising:

recursively querying the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identifying one or more dependencies of specifications of object-oriented code objects stored in the database; and

incorporating the one or more dependencies of specifications of object-oriented code objects into the dependency information tracking array.

67. **(Previously Presented)** The computer-readable medium encoded with logic of Claim 65, further operable to perform the steps comprising:

recursively querying the database for one or more dependencies of implementations of object-oriented code objects in the database;

identifying one or more dependencies of implementations of object-oriented code objects in the database; and

incorporating the one or more dependencies of implementations of object-oriented code objects in the database into the dependency information tracking array.

68. **(Previously Presented)** The computer-readable medium encoded with logic of Claim 65, further operable to perform the steps comprising:

parsing the source code of the database for data manipulation statements that fire triggers; and

identifying one or more data manipulation statements that fire triggers.

recursively querying the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identifying one or more dependencies on triggers of code objects stored in the database; and

incorporating the one or more dependencies on triggers of code objects stored in the database into the dependency information tracking array.

69. **(Previously Presented)** A computer-readable medium encoded with logic operable, when executed on a computer processor, to perform the steps comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database;

recursively querying the database for one or more dependencies of specifications of object-oriented code objects stored in the database;

identifying one or more dependencies of specifications of object-oriented code objects stored in the database;

recursively querying the database for one or more dependencies of implementations of object-oriented code objects in the database;

identifying one or more dependencies of implementations of object-oriented code objects in the database;

parsing the source code of the database for data manipulation statements that fire triggers;

identifying one or more data manipulation statements that fire triggers;

recursively querying the database for one or more dependencies on triggers of code objects stored in the database based on the one or more data manipulation statements that fire triggers;

identifying one or more dependencies on triggers of code objects stored in the database; and

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects stored in the database, one or more dependencies of specifications of object-oriented code objects stored in the database, one or more dependencies of implementations of object-oriented code objects stored in the database, and one or more dependencies on triggers of code objects stored in the database.

70. **(Previously Presented)** The method of Claim 41, wherein the specifications of object-oriented code objects comprise PL/SQL specifications for a collection of stored functions and procedures identified as a single entity.

71. **(Previously Presented)** The method of Claim 42, wherein the implementations of object-oriented code objects comprise PL/SQL implementations for a collection of stored functions and procedures identified as a single entity.

72. **(Previously Presented)** A method of generating dependency information for code objects stored in a database, comprising:

recursively querying a database for one or more dependencies of procedural code objects stored in the database;

identifying one or more dependencies of procedural code objects stored in the database;

generating a dependency information tracking array based on the identification of one or more dependencies of procedural code objects; and

for each of the one or more dependencies identified, determining whether the dependency already occurs in the graph; and

terminating the recursive query of the database upon determining that one of the one or more dependencies already occurs in the graph.

73. **(Previously Presented)** The method of Claim 40, further comprising displaying a dependency graph to a user, the dependency graph generated based at least in part on the dependency information tracking array.

74. **(Previously Presented)** The method of Claim 40, further comprising identifying one or more cyclic dependencies among code objects stored in the database.